

特開平11-73381

(43) 公開日 平成11年(1999) 3月16日

(51) Int.Cl.⁶

G 0 6 F 13/00

識別記号

3 5 1

F I

G 0 6 F 13/00

3 5 1 G

3 5 1 H

G 0 6 T 1/00

15/66

A

審査請求 未請求 請求項の数15 O L (全 20 頁)

(21) 出願番号 特願平9-320775

(22) 出願日 平成9年(1997)11月21日

(31) 優先権主張番号 特願平9-169785

(32) 優先日 平9(1997)6月26日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 松尾 康博

神奈川県海老名市本郷2274番地 富士ゼロ

ックス株式会社海老名事業所内

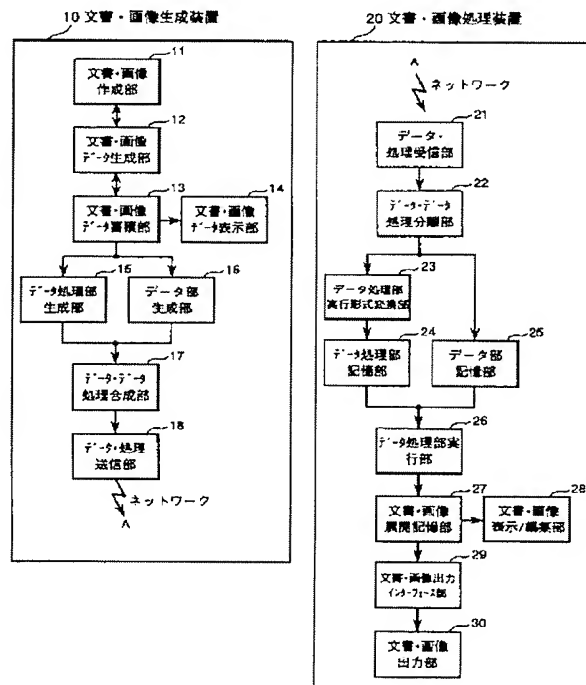
(74) 代理人 弁理士 船橋 國則

(54) 【発明の名称】 画像送信装置、画像受信装置および画像処理システム、並びに文書・画像フォーマット

(57) 【要約】

【課題】 ネットワークコンピューティング環境において、様々なコンピューティングデバイスおよびネットワーク環境に依存しない電子ドキュメントの相互交換・伝送・表示・編集・保存・印刷処理環境を実現する。

【解決手段】 文書・画像生成装置10において、文書・画像データ生成部12で生成した文書・画像データから、データ部をデータ部生成部16で生成しかつデータ処理部をデータ処理部生成部15で生成し、これらをデータ・データ処理合成部17で合成し、データ・処理送信部18から送信する。一方、文書・画像処理装置20において、受信した文書・画像データをデータ・データ処理分離部22でデータ部とデータ処理部とに分離し、データ処理部を取り込んだ環境にて実行し得る文書・画像処理形式にデータ処理部実行形式変換部23で変換し、これに基づいてデータ処理部実行部26でデータ部を解釈・実行する。



【特許請求の範囲】

【請求項1】 文書・画像データを生成し、この文書・画像データに対してその処理プログラムを付加する文書・画像生成手段と、

前記文書・画像生成手段によって生成された文書・画像データおよびこの文書・画像データに対する処理プログラムを送信する送信手段とを備えることを特徴とする画像送信装置。

【請求項2】 前記文書・画像生成手段は、文書・画像データを生成するデータ生成手段と、前記データ生成手段によって生成された文書・画像データとこの文書・画像データを解釈・展開する処理プログラムとを合成して出力する合成手段とを有することを特徴とする請求項1記載の画像送信装置。

【請求項3】 文書・画像データおよび前記文書・画像データに対して付加された処理プログラムを受信する受信手段と、前記受信手段により受信された処理プログラムを実行させるデバイス非依存の実行手段とを備えることを特徴とする画像受信装置。

【請求項4】 前記実行手段は、前記受信手段によって受信された前記文書・画像データと前記処理プログラムとを分離する分離手段と、前記分離手段によって分離された前記処理プログラムを取り込んだ環境にて実行する文書・画像処理形式に変換する実行形式変換手段と、前記実行形式変換手段によって形式変換された前記処理プログラムに基づいて前記分離手段によって分離された前記文書・画像データを解釈・実行するデータ実行手段とを有することを特徴とする請求項3記載の画像受信装置。

【請求項5】 文書・画像データを生成し、この文書・画像データに対してその処理プログラムを付加する文書・画像生成手段と、前記文書・画像生成手段によって生成された文書・画像データおよびこの文書・画像データに対する処理プログラムを送信する送信手段とを有する画像送信装置と、

前記画像送信装置から送信される前記文書・画像データおよびこの文書・画像データに対して付加された前記処理プログラムを受信する受信手段と、前記受信手段によって受信された処理プログラムを実行させるデバイス非依存の実行手段とを有する画像受信装置とを備えることを特徴とする画像処理システム。

【請求項6】 前記文書・画像生成手段は、文書・画像データを生成するデータ生成手段と、前記データ生成手段によって生成された文書・画像データとこの文書・画像データを解釈・展開する処理プログラムとを合成して出力する合成手段とを有することを特徴とする請求項5記載の画像処理システム。

【請求項7】 前記実行手段は、前記受信手段によって受信された前記文書・画像データと前記処理プログラム

とを分離する分離手段と、前記分離手段によって分離された前記処理プログラムを取り込んだ環境にて実行する文書・画像処理形式に変換する実行形式変換手段と、前記実行形式変換手段によって形式変換された前記処理プログラムに基づいて前記分離手段によって分離された前記文書・画像データを解釈・実行するデータ実行手段とを有することを特徴とする請求項5記載の画像処理システム。

【請求項8】 前記文書・画像生成手段は、異なる形式の文書・画像データを生成する複数のデータ生成手段と、前記複数のデータ生成手段によって生成された複数の文書・画像データとこれら複数の文書・画像データを解釈・展開する処理プログラムとを合成して複数の文書・画像ファイルとして出力する第1の合成手段と、前記第1の合成手段から出力される複数の文書・画像ファイルを合成して単一の文書・画像ファイルとして出力する第2の合成手段とを有し、前記送信手段は前記第2の合成手段から出力される単一の文書・画像ファイルを前記画像受信手段との間で相互伝送することを特徴とする請求項5記載の画像処理システム。

【請求項9】 前記受信手段は前記送信手段との間で前記単一の文書・画像ファイルの相互伝送を行い、前記実行手段は、前記受信手段によって受信された前記単一の文書・画像ファイルを各項単位に分離して複数の文書・画像ファイルとして出力する第1の分離手段と、前記第1の分離手段によって分離された前記複数の文書・画像ファイルをそれぞれ前記文書・画像データと前記処理プログラムとを分離する第2の分離手段と、前記第2の分離手段によって分離された前記処理プログラムを取り込んだ環境にて実行する文書・画像処理形式に変換する実行形式変換手段と、前記実行形式変換手段によって形式変換された前記処理プログラムに基づいて前記第1の分離手段によって分離された前記文書・画像データを解釈・実行するデータ実行手段とを有することを特徴とする請求項8記載の画像処理システム。

【請求項10】 前記画像送信装置が複数設けられていることを特徴とする請求項5記載の画像処理システム。

【請求項11】 前記画像送信装置が複数の作成・実行環境を有することを特徴とする請求項5又は10記載の画像処理システム。

【請求項12】 前記実行手段は、前記複数の画像送信装置から送信される複数の文書・画像データを単一形式の文書・画像ファイルとして取り扱い、項順変換および編集処理を実行する操作手段を有することを特徴とする請求項10記載の画像処理システム。

【請求項13】 文書・画像データの構成要素データであるデータ部と前記文書・画像データを解釈・展開するためのデータ処理部とを対有し、前記データ部と前記データ処理部とが分離可能であることを特徴とする文書

・画像フォーマット。

【請求項14】 前記データ部を解釈し、相互交換・伝送・表示・編集・保存・印刷可能な形式に変換するための実行環境を他の実行資源環境に依存することなく、自己凍結形式にて提供することを特徴とする請求項13記載の文書・画像フォーマット。

【請求項15】 前記データ部は、伝送される資源実行環境に対応するカラー固有情報であり、前記データ処理部は、前記実行資源環境において前記データ部を相互交換・伝送・表示・編集・保存・印刷可能な形式に変換実行する色変換処理部であることを特徴とする請求項13記載の文書・画像フォーマット。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークコンピュータ上で作成される電子ドキュメントをどのようなネットワークコンピュータ環境においても相互交換・伝送・表示・編集・保存・印刷可能にする画像処理システムおよび文章・画像フォーマット（電子ドキュメント構造）に関し、特にデバイス非依存な文書・画像実行環境の提供、自立実行・自己凍結型の文書・画像ファイルおよび文書・画像フォーマットの経時可読性を保証するための画像処理システムおよびこれに用いられる文書・画像フォーマットに関するものである。

【0002】

【従来の技術】近年、マルチメディア、コンピュータハードウェア・ソフトウェア、DTPハードウェア・ソフトウェアおよびネットワークインフラ等の技術的進歩により様々な電子ドキュメントが簡単に作成され、相互交換・伝送・表示・編集・保存および印刷可能な環境が整ってきた。このような状況の中、パーソナルコンピュータを代表とするコンピュータ資源にて実行される各種アプリケーションプログラム上で作成された電子ドキュメントを、インターネットにて代表されるネットワーク環境上で相互交換・伝送・表示・編集・保存・印刷するために各種ファイルフォーマット、アプリケーションソフトウェアおよびシステムが出現し用いられている。

【0003】その代表的なファイルフォーマットとしては、相互交換・伝送・表示・編集・保存・印刷等に用いられるAdobe Systems社のPDF（ポータブル・ドキュメント・フォーマット）やKodak社のFlashPixフォーマット、WWW（ワールド・ワイド・ウェブ）環境でコンテンツ情報を伝送・表示する目的にて用いられるHTML（ハイパー・テキスト・マークアップ・ランゲージ）等が挙げられる。

【0004】また、それらを表示・編集するソフトウェアプログラムとして、Adobe Systems社のAcrobat Reader/Writer/Capture、Kodak社のFlashPix Reader/Writer、そしてインターネット上でHTTP

（ハイパー・テキスト・トランスファー・プロトコル）を用い、HTMLで記述した電子ドキュメント（コンテンツ情報）を伝送し表示するために用いられるWWWブラウザソフトウェアとしてNetScape社のNetScape NavigatorやMicrosoft社のInternet Explorerが代表的なものとして挙げられる。

【0005】ここ数年でこのようなものが利用可能なネットワークコンピューティング&電子ドキュメント環境が広がり、インターネットに代表されるネットワーク環境に多くの電子ドキュメントが存在するようになってきた。また同様に、印刷出力のソースデータとして、ネットワーク上に存在する電子ドキュメントが用いられるようになってきたことにより、印刷（プリンティング）環境においても新しい変化が起きつつある。

【0006】従来型の印刷環境では、ホストコンピュータハードウェア・オペレーティングシステムに依存したプリンタドライバを用い、各種ページ記述言語ファイル（PDLファイル）を出力し、決められたネットワークプロトコルにてプリンタ本体にPDLファイルを伝送し、プリンタ本体が送られてきたPDLファイルを解釈し、プリンタの特性に依存したラスターイメージに展開処理してプリント出力を得るという方法が一般的な方法として挙げられる（例えば、特開平6-162008号公報参照）。

【0007】ここで言うページ記述言語の代表的なものとして、Adobe Systems社のPostScript、Xerox社のInterpressおよび富士ゼロックス社のArtなどが挙げられる。また同様に、ネットワークプロトコルとしては、TCP/IP、Apple Computer社のAppleTalk、Novell社のIPXおよびMicrosoft社のNetBEUI等が代表的なものとして挙げられる。

【0008】ここで、上述したこれまでの一般的なプリンティングアーキテクチャにてWWW上に存在するHTMLにて記述された電子ドキュメントをプリントすることを考えてみる。現状、WWWブラウザにて表示されたHTMLの電子ドキュメントをプリント出力する際には、HTMLの電子ドキュメントのコンテンツ情報がネットワーク上のどのような場所に物理的に存在しようとも、WWWブラウザが存在するホストコンピュータ上にそのデータをいったん読み込み、WWWブラウザおよびプリンタドライバによって各種ページ記述言語もしくはプリンタラスターデータに変換処理を行い、目的とするプリンタ本体にそのデータを送り、解釈・展開処理を行ってプリント出力を得る方法が採られている。

【0009】この方法では、HTMLによる電子ドキュメントのコンテンツ情報を読み込み各種のPDLファイルに変換処理を行うために高速なプリント出力処理が行

えないという問題がある。また、HTMLの電子ドキュメントデータはドキュメント記述能力が低いため、WWWサーバー上に存在する殆どのドキュメントオブジェクトは現在そのデバイスに依存した解像度のラスタープリミティブデータとして取り扱われることが多く、またWWW上でのHTMLによるドキュメントの表示において、カラー情報に対する事前WYSIWYG (What You See Is What YouGet)処理が現状では何も行われていないこと等もあり、一般的なアプリケーションにて生成した電子ドキュメントをPDFファイルに変換してプリント出力を得る電子ドキュメントならびにプリンティング方法に比べると、HTMLによる電子ドキュメントプリント環境の品質は大きく劣っているというのが現状である。

【0010】また一方、これまでの一般的なプリンティングアーキテクチャにおいては、ホストコンピュータハードウェア・オペレーティングシステム・ネットワークプロトコルに依存した各種プリンタドライバをサポートする必要がある。しかしながら、インターネット上に構築されたHTML・HTTPを用いるWWW環境においては、複数のホストコンピュータハードウェア・オペレーティングシステム・ネットワークプロトコルに対して依存することなく、単一の環境で様々なネットワークコンピュータが接続可能なため、電子ドキュメントの伝送・表示・保存・印刷を容易に行うことが可能であるという特性を用いて、HTMLで記述され表示された電子ドキュメントをWWWブラウザが存在するホストコンピュータ環境に一時的に送ることなく、直接ネットワーク上に接続された各種プリンタに送信し、HTMLによる電子ドキュメントを受信するプリンタ本体がHTTPを直接解釈することにより電子ドキュメントファイルとして受け取り、HTML電子ドキュメントファイルを解釈・展開して、プリント出力を得るというプリンティングアーキテクチャを持つ製品が最近いくつか出現しつつある。

【0011】このようなプリンティングアーキテクチャをサポートするプリンタコントローラには、HTTPを解釈するネットワークプロトコルインターフェースおよびHTMLを解釈しプリンタデバイスに依存したラスターイメージを生成するためのインタープリターが備えられている。このプリンティングアーキテクチャにおいては、従来型の一般的なプリンティング環境と異なり、ホストコンピュータハードウェア・オペレーティングシステム・ネットワークプロトコルに依存することなくプリンティング環境を構築可能であり、各種ホストコンピュータ・ネットワークプロトコル・PDFファイルに依存したプリンタドライバを複数個サポートする必要はなくなる。

【0012】また、ネットワーク上に存在するHTMLで記述された電子ドキュメントデータを一度WWWブラウザ

ユーザーの存在するホストコンピュータ上に読み込む必要もないため、従来の一般的なプリンティングアーキテクチャに比べると高速に印刷処理を行うことが可能となる。しかし、やはりHTMLを基本とした電子ドキュメントデータは通常のアプリケーションデータを用いた一般的な方法による印刷出力処理に比べるとその印刷品質は劣っていることも現状では事実となっている。このようなこれから一般的になるであろうインターネットプリンティングに関するアーキテクチャの詳細は、Printer Working Group の" Internet Printing Protocol/1.0:Model and Semantics" というドキュメントを参照することにより把握できる。

【0013】また、同じようなことを、電子ドキュメントの相互交換・伝送・表示・編集・保存処理および電子ドキュメントの経時可読性という観点から考えてみる。これまで、ネットワークおよびネットワークコンピューティング環境が特殊な存在として取り扱われていた時代においては、通常の電子ドキュメントの生成・相互交換・伝送・表示・編集・保存・印刷のすべての処理は電子ドキュメントを生成したアプリケーションソフトウェアに依存したものであった。

【0014】一度特定のアプリケーションソフトウェアにて生成された電子ドキュメントを他の実行環境に伝送してから表示・編集・保存・印刷を行う際には必ず、その電子ドキュメントを作成したときと同様のアプリケーションソフトウェアが必要であった。この制限事項がネットワーク環境において自由な電子ドキュメントの交換の妨げとなっていたことは言うまでもない。

【0015】このような問題に対しても、WWWによるインターネット環境およびその環境で使用される電子ドキュメントフォーマットは影響を与えた。WWW環境は、各種コンピュータハードウェア・オペレーティングシステムに依存しない電子ドキュメントの相互交換・伝送・表示・編集・保存・印刷環境をWWWブラウザ・HTML・HTTPを用いてあるレベルにおいて可能とし、その枠組みの中で電子ドキュメントのネットワーク上での相互交換および経時可読性のある一定レベル向上させることに成功している。また、その枠組みの中で使用される電子ドキュメントの相互交換・伝送・表示・編集・保存および印刷処理を目的としてAdobe Systems社によって開発されたPDFファイルを用いるアプリケーションが最近多くなってきている。

【0016】Adobe Systems社は、このファイルフォーマットを電子ドキュメントの作成・相互交換・伝送・表示・編集・保存・印刷およびペーパードキュメントの電子ドキュメント化に用いることができ、かつコンピュータハードウェア・オペレーティングシステム・アプリケーションソフトウェアに依存しない電子ファイルフォーマットとして位置づけ、インターネットを代表とするネットワークおよびネットワークコンピュー

10 Working Group の" Internet Printing Protocol/1.0:Model and Semantics" というドキュメントを参照することにより把握できる。

20 に依存したものであった。

30 インターネット環境およびその環境で使用される電子ドキュメントフォーマットは影響を与えた。WWW環境は、各種コンピュータハードウェア・オペレーティングシステムに依存しない電子ドキュメントの相互交換・伝送・表示・編集・保存・印刷環境をWWWブラウザ・HTML・HTTPを用いてあるレベルにおいて可能とし、その枠組みの中で電子ドキュメントのネットワーク上での相互交換および経時可読性のある一定レベル向上させることに成功している。また、その枠組みの中で使用される電子ドキュメントの相互交換・伝送・表示・編集・保存および印刷処理を目的としてAdobe Systems社によって開発されたPDFファイルを用いるアプリケーションが最近多くなってきている。

40 集・保存および印刷処理を目的としてAdobe Systems社によって開発されたPDFファイルを用いるアプリケーションが最近多くなってきている。

ティング環境において電子ドキュメントの作成・相互交換・伝送・表示・編集・保存・印刷処理に広く利用でき、また半永久的な経時可読性を保証するファイルフォーマットとして推奨し、各種ビジネスを展開してきている。

【0017】また、1993年より米国Sun Microsystems社を中心として開発が開始されているプログラミング言語としてJava言語というものが存在する。このJava言語の開発方針は、ネットワーク上を簡単に伝送・相互交換可能なポータビリティの追求とデバイス非依存な言語体系と実行処理系の提供にある。一般的にJava言語で作成されたアプリケーションソフトウェアはJavaアプレットと呼ばれ、その実態はどのようなデバイスおよび実行環境にも依存しないJava Byte Codeという形で生成・伝送され、実行環境にて実行処理される。

【0018】Java言語で作成されたJavaアプレットおよびJavaアプリケーションは、Java実行環境を持つWWWブラウザアプリケーションソフトウェアおよび一般的な各種アプリケーション・オペレーティングシステム上で解釈（交換）処理され実行される。Java言語の実行処理系として、Java Byte Codeとハードウェア依存実行マシン言語との仲介（解釈・交換）処理を行うJava Byte Code Interpreterと各種オペレーティングシステムで用いられるウインドウシステムのコントロールとのインターフェースを司るAWT (Abstract Window Toolkit) と呼ばれる実行環境が各種ハードウェアプラットフォーム／オペレーティングシステム・ウインドウシステムに対して開発され各社から提供されている。

【0019】このように、Java言語およびJavaアプレット実行環境においては、アプリケーションプログラムおよびその実行環境と各種コンピュータデバイスとの依存性を排除することで、ネットワークコンピューティング環境におけるアプリケーションプログラムの可搬性・実行環境デバイス非依存性および経時実行性を確保するようにしている。

【0020】

【発明が解決しようとする課題】しかしながら、上述した従来技術においては、ネットワークに存在する電子ドキュメントを高品位かつ高速に相互交換・伝送・表示・編集・保存・印刷処理可能であり、経時可読性を半永久的に保証し得る電子ドキュメントフォーマット、プリンティングアーキテクチャとして、以下のような課題が存在する。

【0021】その一つとして、従来技術の電子ドキュメントフォーマット（PDF、HTML、FlashPix等）は、相互交換・伝送・表示・編集・印刷処理を行う際には必ず特定の専用アプリケーション（Acrobat、WWWブラウザ、FlashPix Read

er/Writer等）を必要とし、電子ドキュメントのコンテンツとプログラムとの物理的かつ論理的同時存在性および依存性の制限が存在するため、電子ドキュメントの経時可読性を完全に保証することができない。また、様々なコンピュータデバイスが存在するネットワークコンピュータ環境において、電子ドキュメントを相互交換・伝送・表示・保存・印刷するための電子ドキュメントフォーマットとして要求される経時可搬性の条件を満たしていない。

【0022】理解しやすい具体的な問題として説明を行うと、ネットワークを介在してPDFファイルによる電子ドキュメントを受け取り、表示・編集・保存・印刷するには必ずPDFファイルを表示・編集・保存・印刷するためのアプリケーションソフトウェア（具体的には、Acrobat Reader）が必要であり、そのアプリケーションソフトウェアを持っていなければ表示・編集・印刷することができない。

【0023】また、ネットワーク環境に必要なアプリケーションが存在して無償で取得できる環境にあっても、そのアプリケーションソフトウェアを入手する処理を行わなければならないという問題がある。さらに、ネットワークより無償で取得できる環境を持っていないユーザーは、そのアプリケーションソフトウェアを取得できない。先述した他の電子ドキュメントフォーマットにおいても同様の問題を持ち合わせている。

【0024】他の一つとして、インターネットを代表とするネットワーク環境において、様々なネットワークコンピュータ資源を介在することなく、電子ドキュメントフォーマットを印刷処理する場合、その電子ドキュメントを印刷した際の品質はその電子ドキュメントのフォーマット形式のもつ表現能力および処理系に依存した形となってしまう、印刷出力を目的とした画像処理装置の品質性能を最大限に引き出すことができない場合がある。

【0025】理解しやすい具体的な問題として説明を行うと、HTMLで作成されたWWWブラウザのコンテンツデータを直接画像処理装置に送り処理する印刷処理系において、現状その印刷品質はHTMLの表現形式に依存した形となっており、ほとんどの描画オブジェクトはその表示デバイス（作成環境）に依存したビットマップ形式に近いラスタープリミティブとして処理される。このように様々なネットワークコンピューティングデバイスの依存性を排除し、電子ドキュメントデータを印刷（表示）する系においては、高画質表現を必要とする電子ドキュメントフォーマットとして特定の環境に依存する画像・描画フォーマットしか選択できない場合、高品質な印刷装置（印刷・表示環境）に対しては様々な問題を引き起こす。

【0026】しかし、印刷環境を主体に考え、ドキュメント表現能力の高い単一のフォーマット構造にしてしまうことはドキュメント構造の展開処理の複雑さと全体的

な処理性能の劣化という問題を引き起してしまうこととなる。これと同様の問題は、電子ドキュメントを相互交換・伝送・表示・編集・保存を行う電子ドキュメントの搬送媒体と表示データ構造が独立ではなく、一意に決定依存してしまうネットワークドキュメントシステムおよび電子ドキュメントフォーマットアーキテクチャにおいても起こる。本来ならば、伝送媒体と電子ドキュメントフォーマットは完全に独立性を持たなければならない。またそれらを決定するのはドキュメント作成者であることが望ましい。

【0027】本発明は、上記課題に鑑みてなされたものであり、その目的とするところは、ネットワークコンピューティング環境において様々なコンピューティングデバイスおよびネットワーク環境に依存しない電子ドキュメントの相互交換・伝送・表示・編集・保存・印刷処理環境を実現する画像処理システムおよびこれに用いられる文書・画像データフォーマット（電子ドキュメント構造）を提供することを目的とする。

【0028】

【課題を解決するための手段】請求項1記載の画像送信装置は、文書・画像データを生成し、この文書・画像データに対してその処理プログラムを付加する文書・画像生成手段と、この文書・画像生成手段によって生成された文書・画像データおよびこの文書・画像データに対する処理プログラムを送信する送信手段とを備えている。また、請求項3記載の画像受信装置は、文書・画像データおよびこの文書・画像データに対して付加された処理プログラムを受信する受信手段と、この受信手段により受信された処理プログラムを実行させるデバイス非依存の実行手段とを備えている。そして、請求項5記載の画像処理システムは、請求項3記載の画像送信装置と、請求項5記載の画像受信装置を備えている。

【0029】上記構成の画像処理システムでは、電子ドキュメントのコンテンツとプログラムの構造を、ドキュメントコンテンツデータとそれらを表示・編集するためのドキュメントプログラムにそれぞれ分離して別々に存在させるのではなく、画像送信装置において、電子ドキュメントを作成し、電子ドキュメントデータを生成する際に、ドキュメントコンテンツデータ（文書・画像データ）とドキュメントプログラム（処理プログラム）の両方を対で持つ電子ドキュメントファイルデータとして生成する。

【0030】その際に、ドキュメントプログラムは、ネットワークコンピューティング環境における様々なコンピューティングデバイスに依存しないプログラミング言語および実行環境を持つプログラムデータとして生成される。これにより、画像受信装置（ネットワークコンピューティング環境）においては、どのようなアプリケーションプログラムも必要とすることなく、自立的に自己解凍形式に、即ちデバイスに依存することなく（デバイ

ス非依存）、その電子ドキュメントを表示・編集・保存・印刷処理可能な実行環境を提供し得る。

【0031】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して詳細に説明する。図1は、本発明の第1実施形態に係る画像処理システムの構成の概略を示すブロック図である。

【0032】図1において、本実施形態に係る画像処理システムは、文書・画像データを生成し送信を行う文書・画像生成装置（画像送信装置）10と、この文書・画像生成装置10から送信された文書・画像データを受信しその処理を行う文書・画像処理装置（画像受信装置）20とからなる。文書・画像生成装置10は、文書・画像作成部11、文書・画像データ生成部12、文書・画像データ蓄積部13、文書・画像データ表示部14、データ処理部生成部15、データ部生成部16、データ・データ処理合成部17およびデータ・処理送信部18により構成されている。

【0033】この文書・画像生成装置10において、文書・画像作成部11は、様々な形式の文書・画像をパソコン等のコンピュータ上で作成するためのものである。この文書・画像作成部11によって作成・編集された文書・画像は、文書・画像データ生成部12において文書・画像データとして文書・画像作成部11に依存した文書・画像データ形式に変換されて出力される。ここで言う文書・画像作成部11および文書・画像データ生成部12は、一般的な文書・画像作成アプリケーションソフトウェア（動作環境ハードウェアを含む）として考えることができる。

【0034】一般的に、文書・画像データ生成部12にて生成された文書・画像データは、文書・画像データ蓄積部13によって蓄積され、保持される。また、この文書・画像データ蓄積部13に蓄積された文書・画像データは、文書・画像データ表示部14にて表示されかつ編集処理が行われる。ここで言う文書・画像データ蓄積部13は、コンピュータ上の主記憶メモリであるハードディスクとして考えることができる。また、文書・画像データ表示部14としては、コンピュータのディスプレイまたはその他の表示デバイスでも良い。

【0035】次に、文書・画像データ蓄積部13に蓄積された文書・画像データ、即ち作成・編集が完了した文書・画像データはデータ処理部生成部15にわたされ、文書・画像データを解釈・展開するために必要なデータ処理部、即ち文書・画像データを相互交換・保存・伝送・表示・編集・印刷等の目的にて用いるためのデータ処理部として出力される。同様に、データ部生成部16にもわたされ、文書・画像データの構成要素データであるデータ部として変換されて出力される。

【0036】データ処理部生成部15にて抽出されたデータ処理部およびデータ部生成部16にて抽出されたデ

ータ部は、データ・データ処理合成部17にわたされて一つのあらたな文書・画像ファイルとして生成される。データ・データ処理合成部17にて生成された文書・画像ファイルは、データ・処理送信部18にわたされ、ネットワーク上の様々な場所に接続された他の画像処理装置に配布される。

【0037】一方、文書・画像処理装置20は、データ・処理受信部21、データ・データ処理分離部22、データ処理部実行形式変換部23、データ処理部記憶部24、データ部記憶部25、データ処理部実行部26、文書・画像展開記憶部27、文書・画像表示／編集部28、文書・画像出力インターフェース部30および文書・画像出力部30により構成されている。

【0038】この文書・画像処理装置20において、データ・処理受信部21にて受信された文書・画像データは、データ・データ処理分離部22においてデータ部およびデータ処理部に分解処理される。このデータ・データ処理分離部22にて分解処理されることによって生成されたデータ処理部は、データ処理部実行形式変換部23にて本文書・画像処理装置20のハードウェアおよびソフトウェア実行環境にて実行可能な形式に変換される。ここで出力される実行可能な形式とは、その環境にて実行可能なソフトウェアプログラムまたはハードウェアスクリプトなどを指す。データ処理部実行形式変換部23にて受信環境で実行可能な実行形式に変換されたデータ処理部は、データ処理部記憶部24に一時的に記憶される。

【0039】一方、データ・データ処理分離部22にて分離されることによって抽出されたデータ部はデータ部記憶部25に記憶され、一時的に保持される。データ処理部記憶部24とデータ部記憶部25に記憶された文書・画像データのデータ処理部とデータ部はデータ処理部実行部26にわたされ、目的とする文書・画像データを表示・編集・印刷・保存可能な形式に変換処理される。

【0040】データ処理部実行部26にて実行され、形式変換処理された文書・画像データは文書・画像展開記憶部27にわたされ、編集・表示に際しては文書・画像表示・編集部28によって表示処理が行われる。また、文書・画像展開記憶部27に記憶された文書・画像データがプリント出力を目的としている場合は、文書・画像出力インターフェース部29を経由して文書・画像出力部30に送られることにより、印刷出力を得ることが可能である。

【0041】ここで、上記構成の第1実施形態に係る画像処理システムの構成と従来例に係る画像処理システムの構成の違いについて、図2および図3を用いて説明する。なお、図2は従来例に係る画像処理システムの構成図、図3は第1実施形態に係る画像処理システムの構成図である。なお、図2および図3において、同等部分には同一符号を付して示してある。

【0042】先ず、図2に示す従来例に係る画像処理システムにおいて、ドキュメント作成コンピュータ31上のアプリケーションプログラム34で作成された文書・画像データは、アプリケーションデータ35としてドキュメント作成コンピュータ31上に存在するアプリケーションプログラム34に依存したデータフォーマットにて保存・伝送処理が行われ、ドキュメント受信コンピュータ32に送られる。ドキュメント受信コンピュータ32において、送られてきたアプリケーションデータ35を電子ドキュメント40としてドキュメント受信コンピュータ32上に表示し、編集可能な状態にするには、ドキュメント生成コンピュータ31上で文書・画像作成に用いたアプリケーションプログラム34が必要である。

【0043】また同様に、ドキュメント受信コンピュータ32にて受信したアプリケーションデータ35により文書・画像をプリンター33において印刷処理する際には、印刷すべきプリンター33に対するプリントデータ37を出力するために、アプリケーションプログラム34、プリントデータクリエーター（プリンタードライバ）36を必要とし、また同時に従来例のプリンター33にはプリントサービスプログラム38とプリントプログラム39が必要である。

【0044】これに対し、図3に示す第1実施形態に係る画像処理システムでは、ドキュメント作成コンピュータ31上のアプリケーションプログラム34で作成された文書・画像データは、ドキュメント作成コンピュータ31上に存在するコンポーザー42により、その文書・画像データのデータ部とその処理プログラムであるデータ処理部を併せ持つドキュレット43として生成され、ドキュメント受信コンピュータ32に送られる。

【0045】ドキュレットデータ43を受け取ったドキュメント受信コンピュータ32にて電子ドキュメント40として表示・編集処理を行う際には、従来例とは異なり、文書作成時に使用したドキュメント作成コンピュータ31上に存在するのと同じアプリケーションプログラム34を必要とすることなく、ドキュレット43を展開処理することによって文書・画像データのデータ部とデータ処理部に分離し、データ処理部をドキュレット実行環境44にて実行可能な形式に変換し、その後データ部を読み込み実行処理することにより、電子ドキュメント40を表示・編集可能な実行環境を得ることが可能となる。

【0046】また、印刷処理に際しても同様に、従来例とは異なり、アプリケーションプログラム34、プリントデータクリエーター（プリンタードライバ）36、プリントサービスプログラム38、プリントプログラム39を必要とすることなくプリンター33への印刷出力処理が可能である。この際に、ドキュメント作成コンピュータ31から送られてきたドキュレット43のデータ処理部には、プリントデータクリエーター（プリンター

ドライバー) 36、プリントサービスプログラム38、プリントプログラム39と同様の処理を含む。そして、それらのプログラムをドキュレット実行環境44にて実行可能な形式に変換し、プリントすべきデータ部を読み込み、展開処理およびプリントサービス処理を行うことにより実行される。

【0047】また、第1実施形態においては、ドキュメント生成コンピュータ31、ドキュメント受信コンピュータ32およびプリンター33の間の文書・画像データ形式としてはドキュレット43による単一データフォーマットの伝送が可能となる。また電子ドキュメント40に対する処理およびペーパードキュメント41を取得する印刷処理の両方においても、ドキュレット43はドキュメント生成コンピュータ31、プリンター33において自立的に自己解凍形式にて実行可能であり、ま実行環境としても単一のドキュレット実行環境44を用いることができるという利点がある。

【0048】次に、画像処理構成要素／文書・画像フォーマット／電子・ペーパードキュメントの依存関係について、図4を用いて本実施形態に係る画像処理システムと従来例に係る画像処理システムを対比して説明する。なお、図4において、(a)は従来例に係る画像処理システムの場合、(b)は第1実施形態に係る画像処理システムの場合であり、また図2および図3と同等部分には同一符号を付して示してある。

【0049】まず、従来例に係る画像処理システムにおける依存関係について説明するに、文書・画像データ伝送先のパソコン上で電子ドキュメント32を伝送・表示・編集するために用いるには、通常、その電子ドキュメント40を作成したアプリケーション34と電子ドキュメント40のアプリケーションデータ35を必要とする。また、印刷処理を行い、出力されるべきペーパードキュメント41を得るには、アプリケーションプログラム34、アプリケーションデータ35、プリンターデータクリエーター(プリンタードライバー)36をコンピュータ上に必要とし、またプリンター上にはプリントサービスプログラム38とプリントプログラム39を必要としてしまう。

【0050】次に、同様にして第1実施形態に係る画像処理システムにおける依存関係について説明する。ドキュメント生成コンピュータ上で作成された電子ドキュメントは、コンポーザー42によってドキュレット43として生成され、ネットワーク上の他のコンピュータ上に伝送される。伝送されたドキュメントを電子ドキュメント40およびペーパードキュメント41として出力する際は、ドキュレット実行環境44のみを必要とし、単一のデータフォーマットと実行環境においてそれぞれの処理を実行可能である。ここで言うドキュレットとは、そのプログラム・データ構造を制限するものではなく、伝送媒体としての固有形態の一つである。

【0051】図5に、第1実施形態に係る画像処理構成要素／文書・画像フォーマットの内部構造を示す。先ず、コンポーザーの内部構造(a)について説明すると、コンポーザー42は、電子ドキュメント生成コンピュータ上にて動作し、各種アプリケーションプログラムにて生成された文書・画像データをドキュレット43の構成要素に変換する。

【0052】コンポーザー42の構成要素としては、アプリケーションデータをドキュレット43のデータ部として変換・生成処理を行うデータ変換部42-1と、アプリケーションプログラム実行環境をドキュレット43のデータ処理部として変換・生成処理を行うデータ処理プログラム変換部42-2と、印刷やドキュメント配信ジョブ内容等が記述されたジョブサービスプログラムを生成するジョブサービス変換部42-3と、伝送したデータのセキュリティと配信先でのアクセス認証を行うことが可能なセキュリティプログラムを添付するためのセキュリティ添付部42-4とから構成される。

【0053】ここで、データ処理プログラム変換部42-2にて変換処理されるデータ処理プログラムは、この電子ドキュメントを生成したアプリケーションプログラムとは限らず、コンポーザー内部で独自のファイルフォーマットをサポートしデータ部として生成した場合は、その独自データフォーマットを展開処理する展開プログラムが変換処理されてデータ処理部として添付される。

【0054】また、ジョブサービス変換部42-3とセキュリティ添付部42-4のコンポーザーでの動作はオプションなものである。このジョブサービス変換部42-3は次のように利用される。すなわち、ジョブサービス変換部42-3に、伝送後もしくは伝送前に実行処理される様々な印刷に関連した物理的処理を記述するようにする。これは、複数の異なる実行資源環境において生成された文書・画像フォーマットに対して記述可能である。これにより、ジョブサービス変換部42-3に対する印刷に関連した物理的処理は、送信側もしくは受信側においても実行処理可能となる。

【0055】一方、セキュリティ添付部42-4については、文書・画像フォーマットのデータ処理部に、自動的に添付され情報更新可能な自己実行型のセキュリティープログラムとサービスサービスカウントプログラムを含むようにする。これにより、自立的にセキュリティー・課金処理等の実行が可能となる。

【0056】続いて、ドキュレットの内部構造(b)について説明する。ドキュレット43は、ドキュメント生成コンピュータ上で各種アプリケーションプログラムにて作成された文書・画像データを伝送するために、文書・画像データを含むデータ部43-1と伝送先のコンピュータにおいてドキュレット43のデータ部43-1を展開実行して伝送・表示・編集・印刷・保存可能な形式に変換するデータ処理部43-2により構成される。

【0057】データ処理部43-2は、伝送先のコンピュータ上でドキュレット実行環境44において解釈され、実行可能な形式に変換処理されたのちに展開実行されることとなる。ドキュレット43のデータ処理部43-2の内部構造としては、文書・画像データを表示・編集・印刷・保存可能な形式に変換処理するドキュメントプログラム43-2-1と、印刷やドキュメント配信ジョブ等の処理が記述されたドキュメントジョブサービスプログラム43-2-2と、データのセキュリティと配信先でのアクセス認証を行うことが可能なドキュメントセキュリティプログラム43-2-3により構成される。ここで、ドキュメントジョブサービスプログラム43-2-2とドキュメントセキュリティプログラム43-2-3の添付はオプション的なものである。

【0058】次に、ドキュレット実行環境の内部構造(c)について説明を行う。ドキュレット実行環境には、ドキュレット43の実行プログラムを解釈し、実行環境(ハードウェア・ソフトウェア)にて実行可能な実行形式に変換処理を行う実行プログラム解釈・変換処理部44-1と、ドキュレットプログラムそのものの解釈・変換・実行表示処理等を制御するドキュレット処理部44-2と、各種実行環境の表示ソフトウェアに依存した表示処理・インターフェースを実行するドキュメント入出力デバイス依存処理実行部44-3とから構成される。ここで説明したドキュレット実行環境としては、先述したJava言語プログラムの実行環境を利用することが可能である。その構成に関しては後で述べる。

【0059】図6は、第1実施形態に係る画像処理システムにおける文書・画像フォーマットの変換・処理の流れを示したフロー図である。このフロー図を用いて、第1実施形態に係る画像処理システムの動作および文書・画像フォーマットの変換・処理の手順について以下に説明する。

【0060】まず、ドキュメント生成・送信を行う文書・画像生成装置10側の動作を説明すると、アプリケーションプログラムにて作成されたアプリケーションドキュメントファイル45は、コンポーザー42にてデータ・ドキュメント処理・ジョブサービス・セキュリティプログラムを含む実行ドキュメントファイルであるドキュレットファイル46として生成される。この生成されたドキュレットファイル46は、ネットワークを経由して配信先の各種コンピュータに送信される。

【0061】ここからは、ドキュメント受信・処理を行う文書・画像処理装置20側のコンピュータ内の操作について説明を行う。ネットワークを経由して送られてきたドキュレットファイル46は、受信されたコンピュータ上でファイルとして記憶される。ファイルとして記憶されたドキュレットファイル46は、開封時にドキュレット実行環境44にてセキュリティプログラムの解釈・変換・実行処理が行われ、ドキュメントそのものがネッ

トワーク上のセキュリティチェックポイントに対してドキュメント開封のためのセキュリティ認証を行う。

【0062】セキュリティ認証が許可されたなら、ドキュレットファイル内部のデータ部およびデータ処理部を分離するための分離プログラムが実行される。分離プログラムの実行結果としてアプリケーションにより電子ドキュメントの表示・編集・保存処理の場合は、アプリケーションファイル47とそのアプリケーションプログラム48が生成される。同様に、印刷によるペーパードキュメント出力処理の際は、プリントフォーマット49とデコンポーザープログラム50とジョブサービスプログラム51が生成される。

【0063】受信したドキュレットファイル46をアプリケーションファイルとして表示・編集・保存する際には、ドキュレット実行環境44にてアプリケーションプログラム48が実行され、アプリケーションファイル47が読み込まれることによりアプリケーション環境52が提供される。同様に、ペーパードキュメントを取得する印刷処理の際には、ドキュレット実行環境44にてジョブサービスプログラム51が実行され、その後デコンポーザープログラム50が実行され、プリントフォーマットを読み込みプリントレディーフォーマット53の出力のための展開処理が行われる。それらのプログラム実行結果として、プリントレディーフォーマット53が出力され、ペーパードキュメント41を取得する。

【0064】図7は、第1実施形態に係る画像処理システムのドキュレット実行環境として利用可能なJava実行環境の説明図である。本実施形態に係る画像処理システムおよび文書・画像フォーマットを変換処理する実行プログラム実行環境として、JavaプログラムおよびJava実行環境を利用することが可能である。

【0065】図5で説明したコンポーザー42そのものをJava実行プログラムとして提供することも可能であり、かつドキュレット43のデータ処理部43-2のドキュメントセキュリティプログラム43-2-3、ドキュメントジョブサービスプログラム43-2-2、ドキュメントプログラム43-2-1をJava実行プログラムとして提供することで、ドキュレット実行環境44として一般的に良く知られているJava実行環境を利用することができる。

【0066】Java実行環境は、Javaプログラム61と、Javaプログラム61の入出力処理を行うJava処理系62と、Javaプログラム61として生成されたJava Byte Codeを各種コンピュータ実行環境のCPUにて実行可能な実行プログラムに変換処理を行うバイトコードインタープリター(ByteCode Interpreter)63と、各コンピュータ実行環境に存在する各種、各メーカーのCPU64と、各種OS(Operating System)がサポートしているGUI(Graphical User Interface)ウインドウ環境65と、Javaプログ

10

20

30

40

50

ラム実行結果をGUIウインドウ環境65に表示するためのインターフェース処理を行うAWT (Abstract Window Toolkit) 66とから構成されている。

【0067】上述したように、電子ドキュメントのコンテンツとプログラムの構造を、ドキュメントコンテンツデータとそれらを表示・編集するためのドキュメントプログラムにそれぞれ分離して別々に存在させるのではなく、電子ドキュメントを作成し、電子ドキュメントデータを生成する際に、ドキュメントコンテンツデータとドキュメントプログラムの両方を対で持つ電子ドキュメントファイルデータとして生成するとともに、ドキュメントプログラムをネットワークコンピューティング環境における様々なコンピューティングデバイスに依存しないプログラミング言語および実行環境を持つプログラムデータとして生成することで、伝送されたネットワークコンピューティング環境においては、どのようなアプリケーションプログラムも必要とすることなく、自立的に自己解凍形式にその電子ドキュメントを表示・編集・保存・印刷処理可能な実行環境を提供することができる。

【0068】また、その電子ドキュメントファイルを生

成した生成者（クリエイター）が電子ドキュメントデータ形式とそれらを取り扱うドキュメントデータプログラムとの依存関係を保証すると同時に、自由にその形式を選択することができる。さらに、印刷環境においても、印刷処理に用いる印刷処理データフォーマットおよび印刷プログラムを従来技術のように固定の限られたフォーマット形式およびそのフォーマットに対応したインタープリター（デコンポーザー）プログラムとして持つのではなく、そのドキュメントに応じたドキュメントフォーマット・実行環境にて実行し得るようにしたことで、高速かつ高画質な印刷処理を実行可能な印刷環境を提供できるようになる。

【0069】ここで、ドキュメントコンテンツデータ（データ部）のフォーマットは、アプリケーション文書・画像ファイル、OSドローイングメソッド文書・画像ファイル、PDL文書・画像ファイル、ベクターエンコーディングされた文書・画像ファイル、圧縮されエンコーディングされた文書・画像ファイル、ラスターエンコーディングされた文書・画像ファイル、ラスター文書・画像ファイルなど、どのような形式の文書・画像フォーマットでも良く、また動画データファイルでも良い。

【0070】また、ドキュメントプログラム（データ処理部）は、アプリケーション実行プログラム、OSドローイングメソッド解釈・展開プログラム、PDL解釈・展開プログラム、ベクターエンコーディング解釈・展開プログラム、圧縮・エンコーディングファイル解釈・実行プログラム、ラスターエンコーディングファイル解釈・実行プログラム、ラスターファイル解釈・展開実行プログラムなど、どのようなフォーマットを解釈・展開実行処理するプログラム形式でも良い。ただし、データ部

が動画データファイルの場合には、当然のことながら、動画処理プログラムとなる。

【0071】なお、上記実施形態では、文書・画像フォーマットとして、データ部およびデータ処理部を持たせる構成の場合について説明したが、伝送される様々な実行資源環境に対応し得る色空間などのカラー固有情報と、その実行資源環境においてデータ部を相互交換・伝送・表示・編集・保存・印刷可能な形式に変換実行するために用いる色変換処理部とを持たせることも可能である。これにより、伝送された実行資源環境において、カラー固有情報を取得し、色変換処理部を解釈することによってカラー情報に対する色変換処理を実行することが可能となる。

【0072】また、上記実施形態では、文書・画像フォーマットのデータ処理部を、印刷処理の指定やセキュリティー・課金処理などにも利用するとしたが、その利用法としては種々考えられる。例えば、文書・画像フォーマットのデータ処理部に、伝送後もしくは伝送前に実行処理される様々な印刷に関連した処理を実行し得るネットワーク上に存在する仮想画像処理装置モデルそのもの、またはその場所情報を含むようにする。これにより、どのような環境においても、文書・画像フォーマットのデータ部、データ処理部および仮想画像処理装置モデルを用いて様々な印刷に関連した文書・画像データ編集処理の実行が可能となる。

【0073】さらに、文書・画像フォーマットのデータ部およびデータ処理部にそれぞれ特性情報部を持たせ、それらの特性情報を管理する特性情報管理手段との論理的かつ物理的な位置情報とリンク関係を保持し得るようにすることにより、これら特性情報部、データ部およびデータ処理部を最新の構成に変換する、いわゆる電子ドキュメントのバージョンアップを適宜実現できることになる。

【0074】また、上記実施形態では、文書・画像フォーマットのデータ部の実データおよびデータ処理部のプログラムデータを伝送するとしたが、例えばデータ処理部を伝送可能なネットワーク環境の特定の実行資源環境に存在させることによってデータ部とデータ処理部を物理的に分離し、当該実行資源環境においてデータ部を相互交換・伝送・表示・編集・保存・印刷可能な形式に変換実行処理する際に、例えばアドレス指定によってネットワーク環境を介して逐次読み込んで実行するようにすることも可能である。これによれば、伝送するデータサイズを小さくできる利点がある。

【0075】図8は、本発明の第2実施形態に係る画像処理システムの構成の概略を示すブロック図である。図8において、本実施形態に係る画像処理システムは、文書・画像データを生成し送信を行う文書・画像生成装置（画像送信装置）70と、この文書・画像生成装置70から送信された文書・画像データを受信しその処理を行

う文書・画像処理装置（画像受信装置）80とからなる。

【0076】これら文書・画像生成装置70および文書・画像処理装置80の具体的な構成について以下に説明する。まず、文書・画像生成装置70は、複数（本例では、3個）の文書・画像作成部71-1～71-3、3個の文書・画像データ生成部72-1～72-3、文書・画像データ蓄積部73、文書・画像データ表示部74、データ処理部生成部75、データ部生成部76、データ・データ処理合成部77、文書・画像ファイル合成処理部78およびデータ・処理送信部79により構成されている。

【0077】この文書・画像生成装置70において、3個の文書・画像作成部71-1～71-3は、各々異なる形式の文書・画像をパソコン等のコンピュータ上で作成するためのものである。これら文書・画像作成部71-1～71-3によって作成・編集された各形式の文書・画像は、それぞれ対応する3個の文書・画像データ生成部72-1～72-3において文書・画像データとして3個の文書・画像作成部71-1～71-3の各々に依存した文書・画像データ形式に変換されて出力される。ここで言う3個の文書・画像作成部71-1～71-3および3個の文書・画像データ生成部72は、一般的な文書・画像作成アプリケーションソフトウェア（動作環境ハードウェアを含む）として考えることができる。

【0078】一般的に、文書・画像データ生成部72-1～72-3にて生成された形式の異なる複数の文書・画像データは、文書・画像データ蓄積部73に蓄積され、保持される。また、文書・画像データ蓄積部73に蓄積された複数の文書・画像データは、文書・画像データ表示部74にて表示されかつ編集処理が行われる。ここで言う文書・画像データ蓄積部73は、コンピュータ上の主記憶メモリであるハードディスクとして考えることができる。また、文書・画像データ表示部74としては、コンピュータのディスプレイまたはその他の表示デバイスでも良い。

【0079】次に、文書・画像データ蓄積部73に蓄積された複数の文書・画像データ、即ち作成・編集が完了した複数の文書・画像データはデータ処理部生成部75にわたされ、文書・画像データを解釈・展開するために必要な複数のデータ処理部、即ち文書・画像データを相互交換・保存・伝送・表示・編集・印刷等の目的にて用いるための複数のデータ処理部として出力される。

【0080】同様に、データ部生成部76にもわたされ、複数の文書・画像データの構成要素データである複数のデータ部として変換されて出力される。データ処理部生成部75にて抽出された複数のデータ処理部およびデータ部生成部76にて抽出された複数のデータ部は、データ・データ処理合成部77にわたされて一つのあらたな文書・画像ファイルとして生成される。

【0081】ここで、データ・データ処理合成部77に

て生成された文書・画像ファイルが複数の異なる種類の文書・画像作成部71-1～71-3にて作成された文書・画像ファイルである場合は、複数の文書・画像ファイルを文書・画像ファイル合成処理部78にて単一の文書・画像ファイル形式へのマージ（merge）処理を行う。文書・画像ファイル合成処理部78にて生成された単一の文書・画像ファイルは、文書・画像ファイル相互伝送部79にてネットワークを介して目的とする文書・画像共有・交互交換環境へと伝送される。

【0082】一方、文書・画像処理装置80は、文書・画像ファイル相互伝送部81、文書・画像ファイル分離処理部82、データ・データ処理分離部83、データ処理部実行形式変換部84、データ処理部記憶部85、データ部記憶部86、データ処理部実行部87、文書・画像操作部88、文書・画像展開記憶部89、文書・画像表示／編集部90、文書・画像出力インターフェース部91および文書・画像出力部92により構成されている。

【0083】この文書・画像処理装置80において、文書・画像ファイル相互伝送部81にて受信された複数の異なる画像・ファイル形式を有する単一の文書・画像データ（文書・画像ファイル）は、文書・画像ファイル分離処理部82にて各項単位に複数の文書・画像データに分離される。この分離された複数の文書・画像データは、データ・データ処理分離部83においてデータ部およびデータ処理部に分解処理される。

【0084】このデータ・データ処理分離部83にて分解処理されることによって生成されたデータ処理部は、データ処理部実行形式変換部84にて本文書・画像処理装置80のハードウェアおよびソフトウェア実行環境にて実行可能な形式に変換される。ここで出力される実行可能な形式とは、その環境にて実行可能なソフトウェアプログラムまたはハードウェアスクリプトなどを指す。データ処理部実行形式変換部84にて受信環境で実行可能な実行形式に変換されたデータ処理部は、データ処理部記憶部85に一時的に記憶される。

【0085】一方、データ・データ処理分離部83にて分離されることによって抽出されたデータ部はデータ部記憶部86に記憶され、一時的に保持される。データ処理部記憶部85とデータ部記憶部86に記憶された文書・画像データのデータ処理部とデータ部はデータ処理部実行部87にわたされ、目的とする文書・画像データを表示・編集・印刷・保存可能な形式に変換処理される。ここで、複数の文書・画像データを単一形式の文書・画像ファイルとして取り扱い、事前にらびかえ（プリコレーション）等の処理を実行する際には、文書・画像操作部88にてその処理が行われる。

【0086】データ処理部実行部87にて形式変換処理され、文書・画像操作部88にて操作された文書・画像データは、文書・画像展開記憶部89にわたされ、編集

・表示に際しては文書・画像表示・編集部90によって表示処理が行われる。また、文書・画像展開記憶部89に記憶された文書・画像データがプリント出力を目的としている場合は、文書・画像出力インターフェース部91を経由して文書・画像出力部92に送られることにより、印刷出力を得ることが可能である。

【0087】上述した第2実施形態に係る画像処理システムの構成により、複数の文書・画像作成部71-1～71-3にて作成されたそれぞれ形式が異なる文書・画像データを単一形式の文書・画像ファイルとして処理し、共有・相互交換・保存・表示・編集・印刷可能な環境を提供することができる。

【0088】図9は、従来例のドキュメント構造と第2実施形態に係る画像処理システムで処理されるドキュメント構造との比較図である。ここでは、複数の異なるOS・プラットフォーム環境において異なるアプリケーションプログラムにて作成された異なる種類の複数の文書・画像ファイルを単一の文書・画像ファイルとして取り扱う際のドキュメント構造について述べる。図9(a)は、3つの異なるアプリケーションプログラムにて作成されたドキュメントを示しており、それぞれ異なるアプリケーションプログラムにて作成された異なる実行環境・データ形式を持つ文書・画像ファイルである。

【0089】図9(b)は、3つの異なるアプリケーションプログラムにて作成されたドキュメントをOLE(Object Linking and Embedding)技術でマージしたときのドキュメント構造を示しており、アプリケーション3にて作成された文書・画像ファイル(データ)の中にオブジェクトとしてアプリケーション1および2にて作成された文書・画像ファイル(データ)を埋め込んだ構造となっている。この構造にて電子文書・画像ファイルの共有・相互交換・保存・表示・編集・印刷処理を行う場合には、アプリケーション3以外にアプリケーション1および2がアプリケーション3とは全く独立に存在しなければならない。

【0090】図9(c)、は3つの異なるアプリケーションプログラムにて作成されたドキュメントを単一データ形式のアプリケーションでマージしたときのドキュメント構造を示しており、3つの異なるアプリケーション1、2、3で作成された文書・画像ファイルを単一形式のデータ構造(ラスターデータやPDF)にエクスポート(変換)処理することにより、変換後のデータは単一のアプリケーション4で合成・分割処理が可能な形式となる。しかし、これら文書・画像ファイルを共有・相互交換・保存・表示・編集・印刷するには、アプリケーションプログラム4が存在しなければならない。

【0091】これに対し、図9(d)は、3つの異なるアプリケーションプログラムにて作成されたドキュメントを第2実施形態に係る文書・画像生成装置70においてマージ処理したときのドキュメント構造を示したもの

である。第2実施形態に係る文書・画像生成装置70にてマージ処理されたアプリケーションプログラム1、アプリケーションプログラム2およびアプリケーションプログラム3の3つの異なる種類の文書・画像ファイルは、ドキュレット1、ドキュレット2およびドキュレット3として変換処理され、かつドキュレット4としてマージされ、単一形式の文書・画像ファイルを形成する。

【0092】この場合は、それぞれの文書・画像データにアクセスし、マージ・ディバイド処理する際にも、また文書・画像ファイルの共有・相互交換・保存・表示・編集・印刷を行う際にも、他のアプリケーションプログラムに依存することなく、それぞれの処理を実行することが可能となる。

【0093】図10は、第2実施形態に係る画像処理システムにおける文書・画像フォーマットの変換・処理の流れを示すフロー図である。このフロー図を基に、第2実施形態に係る画像処理システムの動作および文書画像フォーマットの変換・処理の流れを説明する。

【0094】まず、文書・画像生成装置(ドキュメント生成・送信サイド)70の動作を説明すると、異なる複数のアプリケーションプログラムにて作成されたアプリケーションプログラム・ファイルA、アプリケーションプログラム・ファイルB、アプリケーションプログラム・ファイルCは、コンポーザー101にてデータ・ドキュメント処理・ジョブサービス・セキュリティプログラムを含む実行ドキュメントファイルであるドキュレットファイルA、ドキュレットファイルB、ドキュレットファイルCとして生成される。生成されたドキュレットファイルA/B/Cは、ドキュレットマージ処理部102にて単一形式の文書・画像ファイルにマージ処理されてドキュレットファイルDとして生成され、その後ネットワークを経由して共有・配信先の各種コンピューターに送信される。

【0095】ここからは、文書・画像処理装置(ドキュメント共有・処理サイド)80のコンピューター内の操作について説明する。文書・画像生成装置70からネットワークを経由して送られてきたドキュレットファイルDは、ドキュレットディバイド処理部103にて分割され、ドキュレットファイルA、ドキュレットファイルB、ドキュレットファイルCとして生成される。これらドキュレットファイルA/B/Cは、ドキュレット実行環境104にて実行処理され、アプリケーションプログラムにて表示処理するにはドキュメントA/B/Cとして生成され、また印刷処理するにはプリントレディフォーマットA/B/Cとして変換処理される。

【0096】変換処理されたドキュメントA/B/CおよびプリントレディフォーマットA/B/Cは、それぞれアプリケーション環境105に読み込まれ、電子ドキュメントA/B/Cとして表示・編集・保存可能な形式となり、またプリントレディフォーマット106に変

換された後、ペーパードキュメントA/B/Cとしてその出力を得ることが可能となる。

【0097】以上説明した第2実施形態においては、文書・画像生成装置70が単一で、この単一の文書・画像生成装置70の中にそれぞれ異なる形式の文書・画像を作成する例えば3個の文書・画像作成部71-1〜71-3を備えた構成の画像処理システム、即ち同じ環境の中でアプリケーションが違う構成の画像処理システムに適用した場合について説明したが、文書・画像生成装置70は必ずしも単一である必要はない。

【0098】例えば、図11に示すように、単一の文書・画像処理装置80に対して、複数の異なるコンピューティング・アーキテクチャからなる文書・画像生成装置70-1〜70-nを設けた構成の画像処理システム、即ち環境(Macintosh, Unix, Windows)のそれぞれのOSが動作するコンピューティング・アーキテクチャが違う構成の画像処理システムとすることも可能である。かかる構成の画像処理システムの場合には、図8に示す文書・画像処理装置80において、文書・画像操作部88が有用な作用をなす。すなわち、複数の文書・画像生成装置70-1〜70-nから伝送される複数の文書・画像データを単一形式の文書・画像ファイルとして取り扱い、項順変換および編集等の処理を実行する。

【0099】このように、単一の文書・画像処理装置80に対して、複数の文書・画像生成装置70-1〜70-nを設けた構成とすることにより、複数の文書・画像生成装置70-1〜70-nにおける各々形式が異なる複数の文書・画像データを、単一の文書・画像フォーマットに変換することなくそれぞれ独立した異なる形式の文書・画像ファイルのまま単一の文書・画像ファイルとして取り扱うことができる。換言すれば、複数の異なるハードウェア・オペレーティング環境の複数の異なるアプリケーションにて作成された複数の電子ドキュメントを単一の電子ドキュメントとして取り扱い、相互交換・伝送・表示・編集・保存・印刷処理を行うことができる。

【0100】

【発明の効果】以上詳細に説明したように、本発明によれば、ネットワークコンピューティング環境において、様々なコンピュータデバイスに依存することなく、電子ドキュメントの相互交換・伝送・表示・編集・保存・印刷処理を行うことが可能なネットワーク・電子ドキュメント環境を提供でき、しかも電子ドキュメントのコンテンツとプログラミングの論理的かつ物理的な同時存在性および論理的依存性の持つ各種制限を排除することができるため、電子ドキュメントの経時可読性を保証できる。

【0101】また、電子ドキュメントの伝送媒体・データ構造・データ処理プログラムを縦続依存関係なく選択可能であるため、電子ドキュメント・ペーパードキュメントを様々なデバイスに出力する際の表示品質等の問題もドキュメント生成サイドにて解決して保証することができる。

【図面の簡単な説明】

【図1】 本発明の第1実施形態に係る画像処理システムの構成の概略を示すブロック図である。

10 【図2】 従来例に係る画像処理システムの構成図である。

【図3】 第1実施形態に係る画像処理システムの構成図である。

【図4】 画像処理構成要素/文書・画像フォーマット/電子・ペーパードキュメントの依存関係について、従来例に係る画像処理システム(a)と第1実施形態に係る画像処理システム(b)とを対比する図である。

20 【図5】 第1実施形態に係る画像処理構成要素/文書・画像フォーマットの内部構造を示す図であり、(a)はコンポーザーの内部構造を、(b)はドキュレットの内部構造を、(c)はドキュレット実行環境の内部構造をそれぞれ示している。

【図6】 第1実施形態に係る画像処理システムにおける文書・画像フォーマットの変換・処理の流れを示すフロー図である。

【図7】 第1実施形態に係る画像処理システムのドキュレット実行環境として利用可能なJava実行環境の説明図である。

30 【図8】 本発明の第2実施形態に係る画像処理システムの構成の概略を示すブロック図である。

【図9】 従来例のドキュメント構造と第2実施形態に係るドキュメント構造の比較図である。

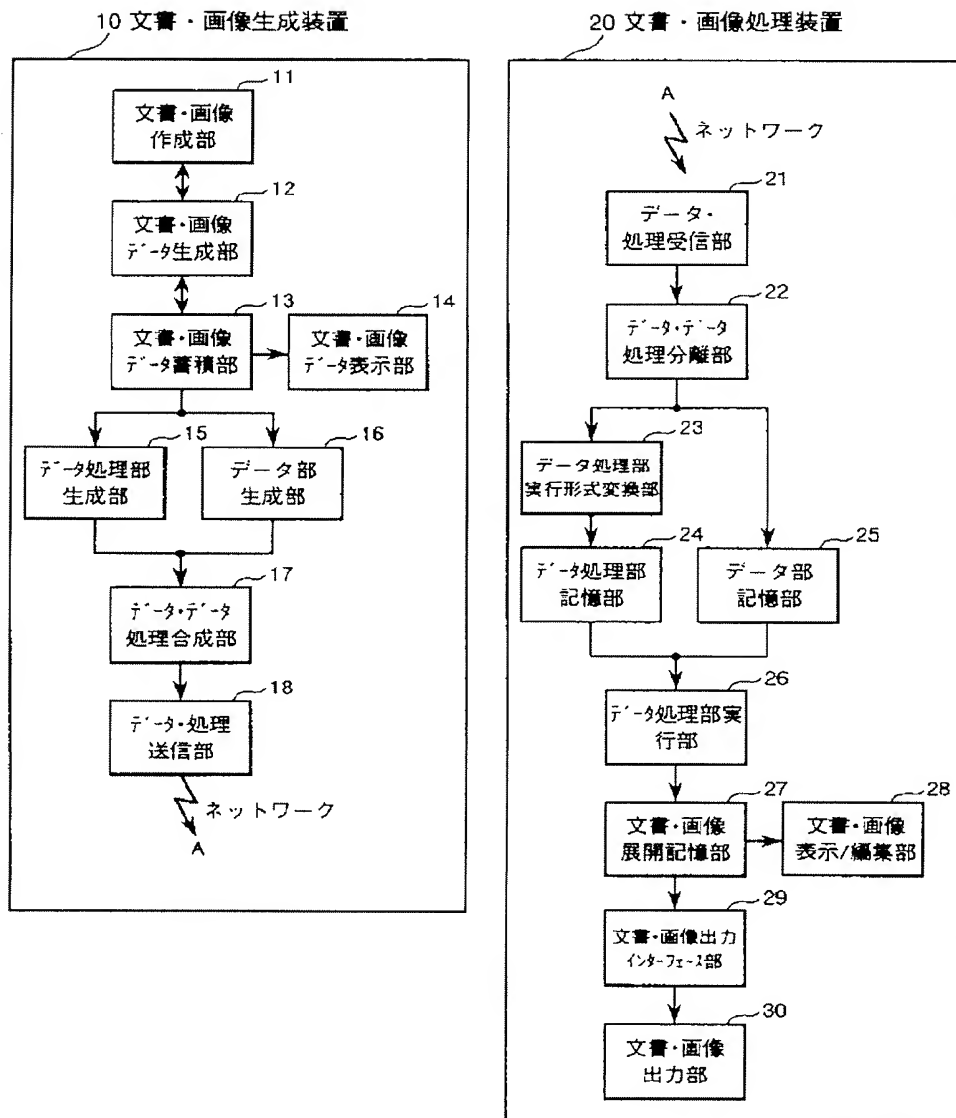
【図10】 第2実施形態に係る画像処理システムにおける文書・画像フォーマットの変換・処理の流れを示すフロー図である。

【図11】 第2実施形態の変形例を示すブロック図である。

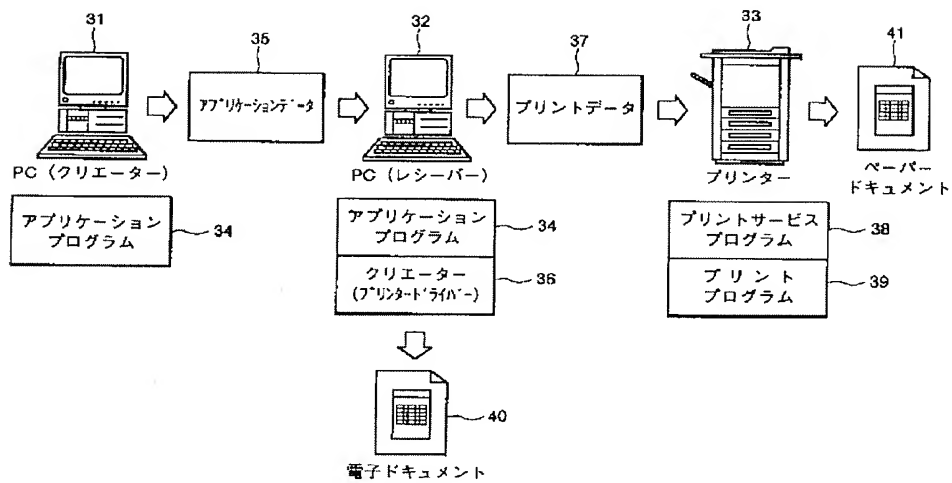
【符号の説明】

10, 70, 70-1〜70-n…文書・画像生成装置、11, 71-1〜71-3…文書・画像作成部、12, 72-1〜72-3…文書・画像データ生成部、15, 75…データ処理部生成部、16, 76…データ部生成部、17, 77…データ・データ処理合成部、20, 80…文書・画像処理装置、22, 83…データ・データ処理分離部、23, 84…データ処理部実行形式変換部、26, 87…データ処理部実行部、78…文書・画像ファイル合成処理部、82…文書・画像ファイル分離処理部、88…文書・画像操作部

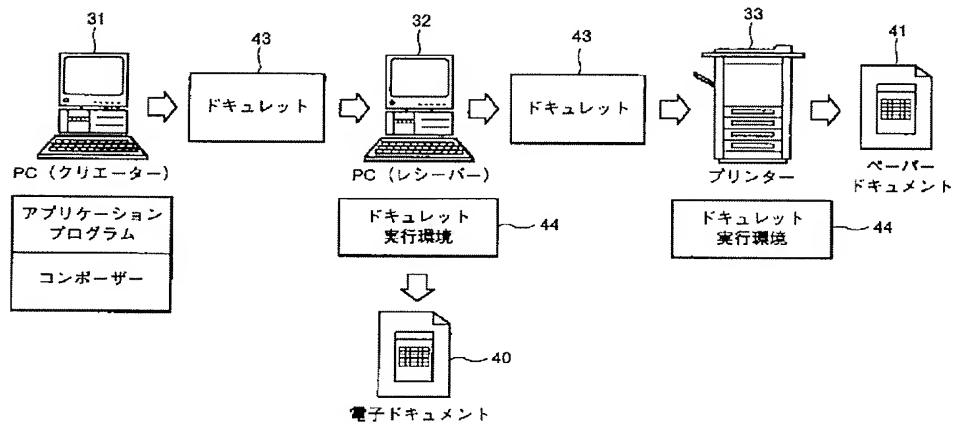
【図1】



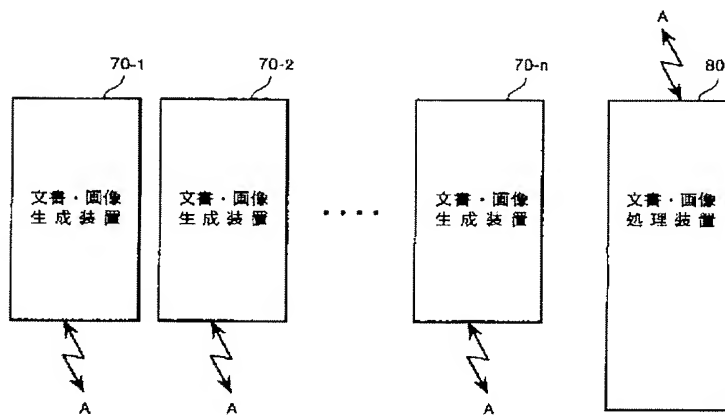
【図2】



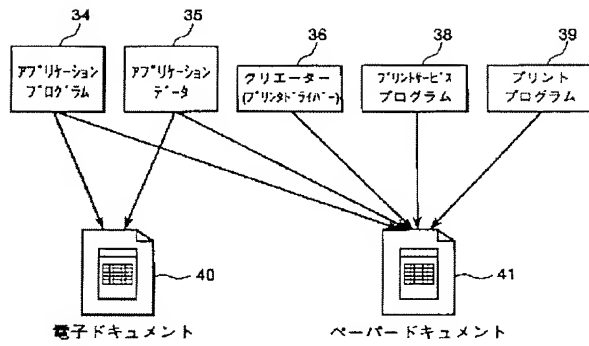
【図3】



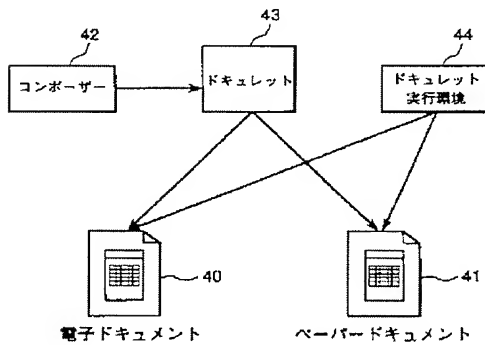
【図11】



【図4】

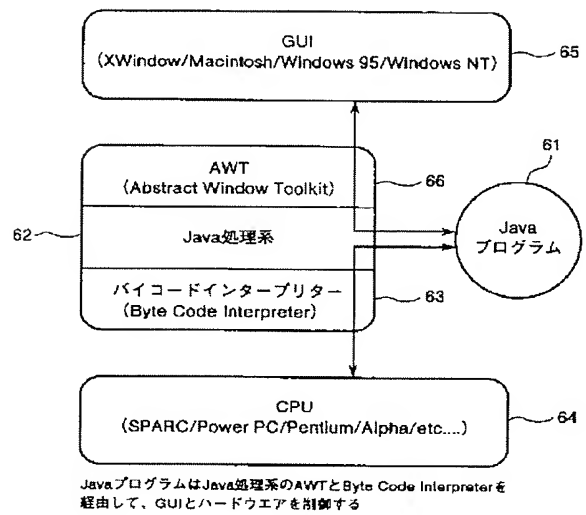


(a)

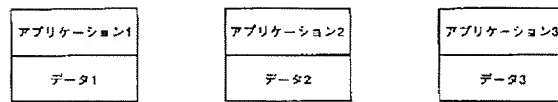


(b)

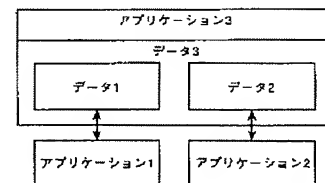
【図7】



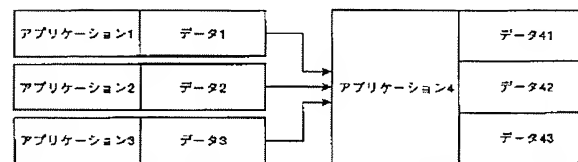
【図9】



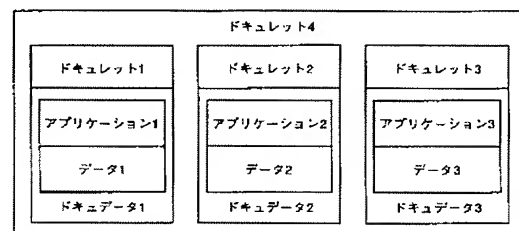
(a)



(b)

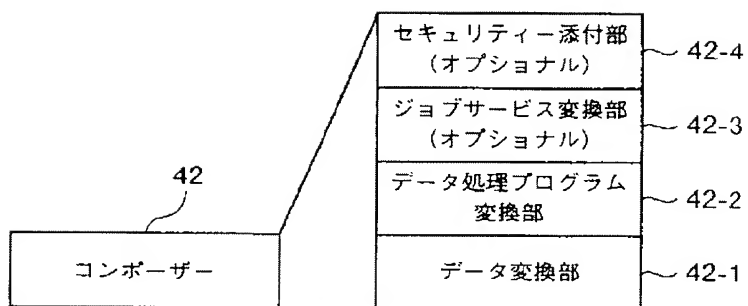


(c)

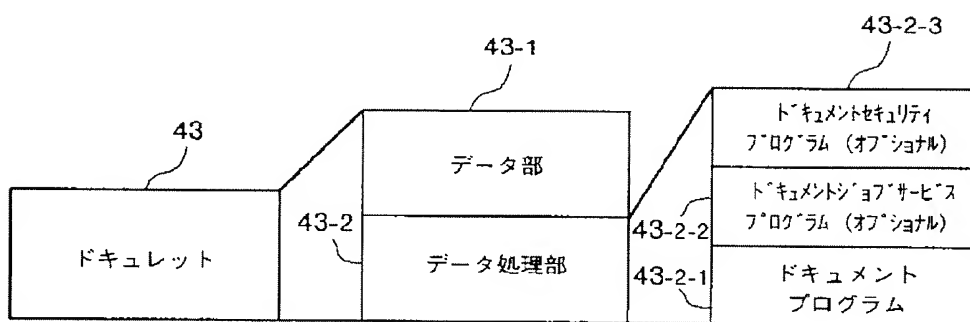


(d)

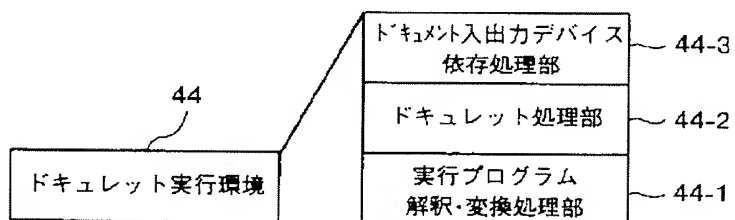
【図5】



(a)

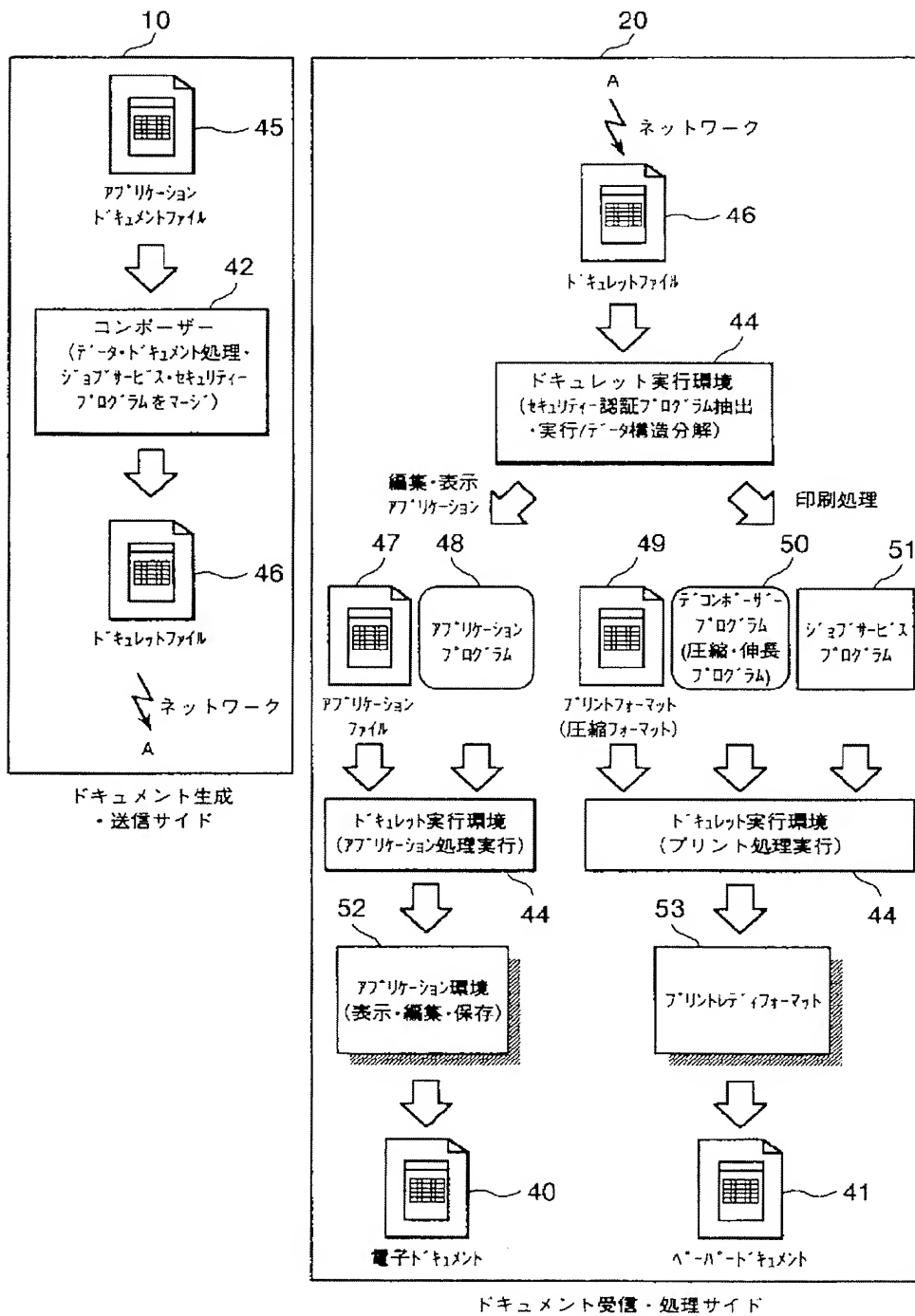


(b)

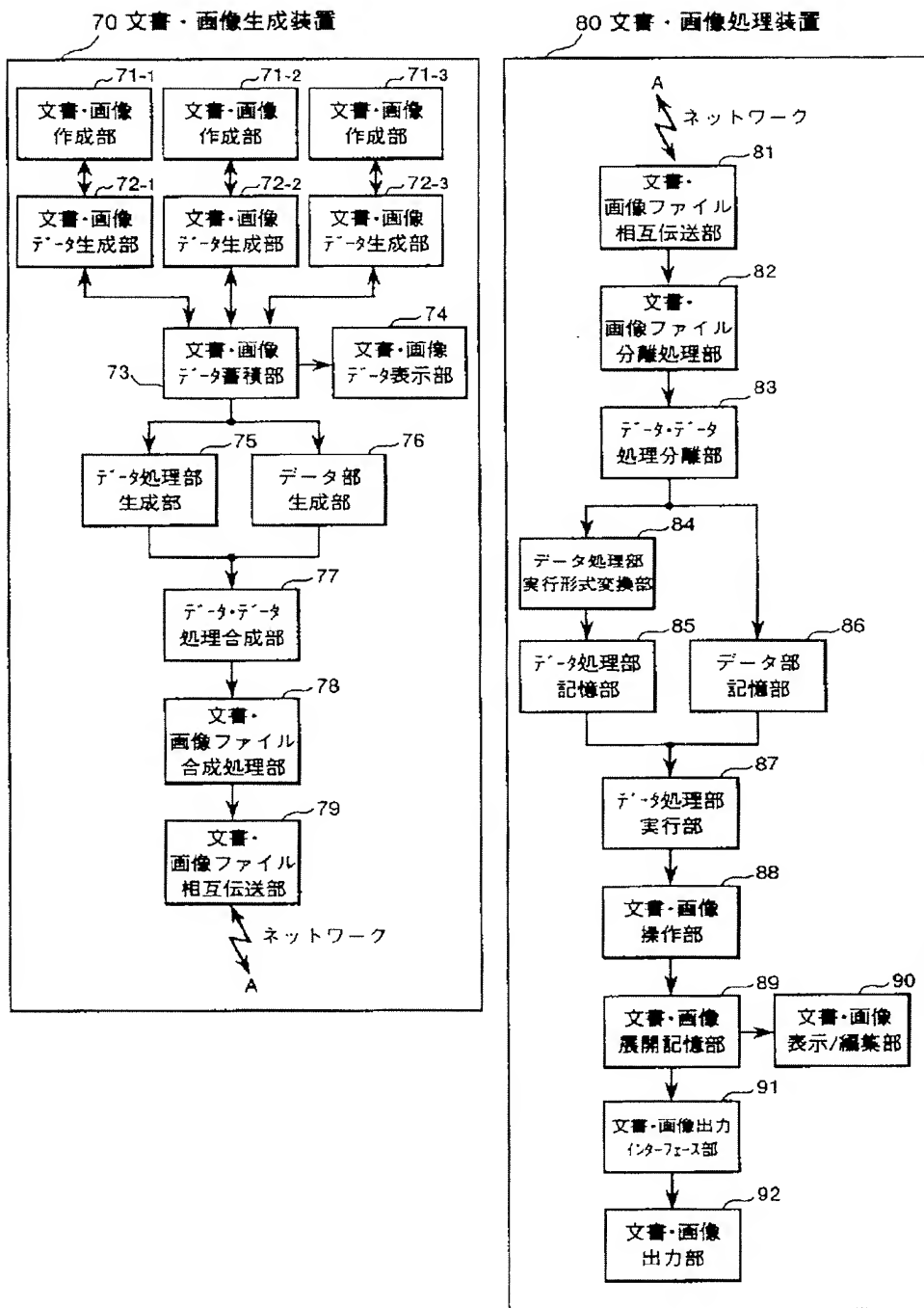


(c)

【図6】



【図8】



【図10】

